IBM Sterling B2B Integrator

IBM

Business Process Modeling Best Practices

Release 5.2

IBM Sterling B2B Integrator

Business Process Modeling Best Practices

Release 5.2

> **Note**
> Before using this information and the product it supports, read the information in "Notices" on page 9.

Contents

Business Process Modeling Best Practices

The business process modeling best practices documentation is designed to provide information that can be a helpful reference when modeling your business processes.

These best practices are especially useful in the following situations:
- While planning your business process models' structure
- While modifying existing business process models
- While troubleshooting problems caused by the structure and configuration of your process models

The documentation assumes that you are generally familiar with terms used in relation to working with IBM® Sterling B2B Integrator. However, if you come across an unfamiliar term, search the documentation library or check the glossary.

Design Guidelines for Business Process Models

Guidelines for the design structure of your business process models relate to the following:
- Naming Conventions
- Reusability
- Structure
- Ongoing Management Strategies

For information about design concepts recommended to ensure more efficient processing, see "Modeling Strategies for Efficient Processing" on page 3.

Using Your Company's Naming Conventions Consistently

Naming different versions of the same business process model brings special considerations. For information, see "Naming Versions Meaningfully" on page 2.

Retaining Service Names in Graphical Process Models

Although you can change the names of services in your graphical process models to make them more descriptive, a good practice is to retain the service name within your new name, so that for future reference you can easily see which service you used for that step (changing the name of a service in the GPM changes the display name shown in the workspace for that service, but it does not rename the service configuration used in the process model).

Retaining the name of the service is particularly helpful so that when you need to update multiple business process models to use a new service (such as when one that you have been using is retired), you can see the service name in the process model. This saves research time.

Designing Reusable Business Processes

Whenever possible, create business processes that you can reuse so that you can avoid the work involved in creating new ones and maintain consistency in your process models.

Using Predefined Business Processes

Whenever possible, use predefined process models provided with Sterling B2B Integrator, to save work.

Configuring Bootstrap Adapters Outside Business Process Models

Sterling B2B Integrator includes several bootstrap adapters – adapters that can start processes dynamically without being part of the process model. When building your business process models, do not include these adapters as the first step in a business process model. Using bootstrap adapters in this way is preferable because your goal is to reuse components whenever possible, for easier maintenance of your process models. If you create an adapter configuration specific to the business process, you are less likely to be able to reuse the adapter configuration to start other business processes.

Keeping Subprocesses Manageable

Keep each subprocess to a manageable size. If the subprocess performs multiple activities, consider splitting it into multiple, smaller subprocesses. This approach enhances reusability and simplifies maintenance.

Another consideration is how you intend to invoke subprocesses. If you are using inline invocation of subprocesses, the impact on performance is relatively small, whereas asynchronous invokes can have a more marked negative effect on performance. If you need to use asynchronous invokes for your subprocesses, consider building the steps into the parent process instead.

Making Modifications Incrementally

Make incremental modifications to your business processes, and thoroughly test each change before making additional changes. This approach can save you a great deal of time because incremental changes are easier to troubleshoot than an entire business process.

Naming Versions Meaningfully

Sterling B2B Integrator stores each version of a business process model that you check in, which enables you to revert to using a previous version should the need arise. Therefore, you should employ a strategy for using the description field to differentiate business process models as you check them in to Sterling B2B Integrator, so that you can easily identify a previous version. For example, you might add a version number or identify the change made to the process

Using the Lock Service

Use the lock service when you have a business process or resource within a business process that should never have more than one instance running at a time. The Lock service will need to be included twice in the business process. The first instance to lock the resource and the second instance to unlock the resource. Generally the lock duration is set to 2-3 times what the estimated time of execution will be so that the lock does not time-out before the business process reaches the unlock step.

Modeling Strategies for Efficient Processing

You can avoid many potential performance problem by implementing well-designed business process models. Optimum performance of your business processes in Sterling B2B Integrator relates to the following:

- Using Appropriate Components
- Workload Management

For information specific to large file handling, see "Modeling Guidelines for Large File Handling" on page 6.

Using the Most Current Services and Adapters

Use the most current services or adapters available in Sterling B2B Integrator. Many newer services replace the activities of more than one older adapter or service, enabling you to reduce the number of steps in a business process, increasing processing efficiency. You can reduce calls to the database, freeing system resources for other processing. The list of retiring services and adapters is a quick way to determine the services and adapters to phase out of your process models. You can also check the Retiring stencil in the GPM.

Using the BPML All Activity Properly

The BPML activity, All (represented in the GPM by the All Start and All End icons), runs two or more complex child activities within a business process simultaneously. Because there are performance implications to using the All activity, use it only if you expect branches of your business process model to take significant amounts of time (in seconds or minutes) to complete (very few services take this long).

Do not use the All activity to run two or more instances of the same subprocess simultaneously. The system will handle the proper load balancing.

Limiting Steps in a Process Model

Do not include excess steps or activities in a process model, such as loops and redundant calls to the database for information that is retained in the process data. Keep the number of service and adapter configurations in the process to a minimum. Excess activities require more system resources. Fewer steps mean shorter processing times.

Using Schedules to Manage Workload

Use scheduling features to manage workloads and improve processing times by moving lower priority processing to off-peak processing times. Scheduling your lower-priority business processes to run during non-peak times frees resources during peak processing times, improving system performance and processing times.

For example, if you have a business process that is low priority and takes many resources, you can schedule the business process to run during your slowest processing time. By doing this, you reduce resource usage during your peak time and improve the processing time of the large, low priority business process.

You can further refine your schedules by excluding busy times and days of the week, or month. For example, you can schedule processes to not run on days you process sales orders in your ERP system.

Be sure to schedule processes to run only as often as necessary to meet your business requirements.

Running Batch Transactions Whenever Appropriate

Run batch transactions instead of individual transactions whenever appropriate. Running batch transactions reduces the number of calls to the database and completes more processing with fewer adapter and service invocations. You can use this technique to improve processing times and manage workload, by reducing processing during peak processing times and maximizing processing opportunities during non-peak processing times. For example:

* You can have a purchase order business process that sends all final purchase orders to a file directory, and a business process that picks up all of the purchase orders from the file directory for processing. If each purchase order is processed when it is received in the file directory, several calls to the database may be made and many invocations made that use system resources. This becomes especially important during peak processing times.

  If, however, you set up a business process to check the folder during non-peak processing hours and process all of the purchase orders at one time, this reduces the use of resources during peak processing and uses your system resources more efficiently during a time that resources are plentiful.

* When you are configuring a business process model to translate and envelope documents, use Deferred mode for the enveloping service whenever your business needs allow. This way, the service translates and envelopes documents in batches at specified intervals.

Using Persistence Levels Appropriately

Associate appropriate persistence levels with the business process and each activity in the business process. Do not use Full persistence unless your business needs require it, as this requires more system resources. Use the persistence level of None if your business process does not require data to be accessed further in the process, and you do not need detailed tracking information.

You may want to use Full persistence when you are testing your processing, to keep more information for troubleshooting scenarios. For example, if you have an outbound EDI business process that has an X12 Envelope activity, the business process data is translated into EDI and the envelope is applied to the message. When the envelope is applied, Sterling B2B Integrator creates a control number for the document. This control number is unique for each document. If you run this business process several times, each document receives a different control number, so you want to persist the enveloping activity in the business process. If the information called by business processes is static (never changing) in the database, you do not need to persist the selectInfo step in a business process, since you can call the same information from the database at any time.

Using Wait and Sleep Services Appropriately

The Sleep service holds onto its processing thread (using more system resources), while the Wait service does not. As a general rule, use the Wait service instead of the Sleep service for wait times longer than one minute, but refer to the following information for the applicability of each service to your business needs:

* For short periods, the Sleep service may be more efficient, because in this case, the system does not need to requeue the process.
* Using a Wait service is likely to make more system resources available.

- A large number of "sleeping" business processes can exhaust resources, resulting in Sterling B2B Integrator being unable to execute more processes. Thus, ensuring that very few processes are sleeping at any one time, and that sleep periods are limited, is critical.
- The Sleep service enables you to set an exact length of time for the process to sleep. The Wait service is far less precise; depending on your Sterling B2B Integrator configuration, the Wait service may wait an extra 10 minutes. If timing is critical, use the Sleep service.

Obtaining Process Information in onFault Layer

If you only need process information in the case of a fault in the process flow (where you have an onFault activity in the process model), add a process information service such as the Get Document Info service or the BP Metadata Info service to the onFault layer (subflow), rather than to the main flow of the business process model.

Testing for Execution Speed and Modify Process Models as Needed

In creating your process models, address speed of execution last. Identify which subprocesses can run in parallel or asynchronously from other subprocesses, and make those changes after your tests confirm that the subprocess or business process is working.

Using Inline Invoke when Possible

Unless you have a business need for an invoked business process to run as a separate process, use the inline invoke option. Processes invoked using the inline option run synchronously as part of the original process, saving the processing time of starting a new unique process.

Invoking Subprocesses Asynchronously

Invoke subprocesses asynchronously when running them in parallel with the main process makes sense, such as in a circumstance in which you need activities to be completed but you do not need to keep track of them from the main process (this is sometimes called *fire and forget*). Unlike using the BPML All activity, in this case, the activities do not synchronize as they do at the end of the All activity, and changes made to process data in the subprocesses are not available in process data.

When you run asynchronous processes, there is no communication back to the parent process unless you include Produce and Consume activities in your process model.

Using Specified Path in XPath Searches

When making an XPath reference to data, using a specified path is more efficient than a "//" search, and the potential difference in processing efficiency can be marked. Although using the "//" search will work, it requires more system resources to complete the search.

For example, because a "//" search finds every possible node that matches the selection criteria, looking for nodes called 'PrimaryDocument' returns every element (regardless of how deeply nested it is) called process data. This extensive search requires every element in process data to be examined. If the search is done within a loop, processing can become significantly slower as the size of process data increases.

Limiting Calls to the File System

If you need repeated access to data from a document on the file system, adding the data from the document to process data can be the more efficient method. Once in process data, the information can be referenced throughout the life of the process.

Modeling Guidelines for Large File Handling

To ensure efficient processing of large files, you must configure your business process models to prevent unnecessary overloading of the database, which can lead to production down scenarios.

Using the For Each Document service along with an Invoke Business Process service in a loop can be a source of production down issues when handling files of any size, but particularly for large files.

A considerable number of documents repeating through the For Each Document service, followed by an Invoke Business Process service, which carries forward all parent documents and process data, causes systems to become overwhelmed by data volume that grows on each cycle through the loop.

Because each child process has a copy of the parent process' original document, and there are N documents to cycle through the loop, N times the amount of documents carry forward. For very large documents or a large number of documents, this can fill the database.

To avoid this problem, structure your business process model as indicated in one of the two methods described here:

Method 1

In your business process model, include:
- An Assign activity with a message_to_child element that selectively passes the correct document (and any other selected data) to the next cycle in the loop
- A Release Service placed to clear out unneeded document copies (do not place the Release service between the Assign activity to message_to_child and the Invoke Business Process service, because in that configuration it will remove the desirable data)

Method 2

In your business process model, include one or more assign elements to message_to_child in the Message To Service of the Invoke Business Process service, to specify the correct document as an output element. You do not need to use the Release service if you use this configuration.

Note: The Sterling B2B Integrator BPML validation check on business process models will report an error for improper document loop configuration. However, if you are using business process models configured with this document loop that pre-date the validation check, and you have not checked the process models out and back in again, you can continue to run them. Recommended practice, however, is to modify their structure to ensure optimum processing based on the instructions provided here.

Example 1 – Properly Configured BPML

The following example format shows the proper configuration of a business
process model that uses method 1:

```
<process name="BPN_TopScopeWithRelease">
  <sequence name="Begin">
    <sequence name="foreachSplitDoc">
      <operation name="For Each Document">
        <participant name="ForEachDocument"/>
        <output message="ForEachDocumentTypeInputMessage">
          <assign to="." from="*"></assign>
        </output>
        <input message="inmsg">
          <assign to="." from="*"></assign>
        </input>
      </operation>

      <assign name="Assign" to="message_to_child" from="PrimaryDocument"></assign>
      <operation name="Invoke Business Process Service">
        <participant name="InvokeBusinessProcessService"/>
        <output message="InvokeBusinessProcessServiceTypeInputMessage">
          <assign to="." from="*"></assign>
        </output>
        <input message="inmsg">
          <assign to="." from="*"></assign>
        </input>
      </operation>

      <operation name="release">
        <participant name="ReleaseService"/>
        <output message="Xout">
          <assign to="TARGET" from="string('message_to_child')"/>
        </output>
        <input message="Xin"/>
      </operation>

      <repeat name="repeater" ref="foreachSplitDoc"/>

    </sequence>
  </sequence>
</process>
```

Example 2 – Improperly Configured BPML

The following example shows a format that will fail validation:

```
process name="BPN_TopScopeNoRelease">
  <sequence name="Begin">
    <sequence name="foreachSplitDoc">
      <operation name="For Each Document">
        <participant name="ForEachDocument"/>
        <output message="ForEachDocumentTypeInputMessage">
          <assign to="." from="*"></assign>
        </output>
        <input message="inmsg">
          <assign to="." from="*"></assign>
        </input>
      </operation>

      <assign name="Assign" to="message_to_child" from="PrimaryDocument"></assign>
      <operation name="Invoke Business Process Service">
        <participant name="InvokeBusinessProcessService"/>
        <output message="InvokeBusinessProcessServiceTypeInputMessage">
          <assign to="." from="*"></assign>
        </output>
        <input message="inmsg">
          <assign to="." from="*"></assign>
        </input>
      </operation>
```

```
                        </operation>

                        <repeat name="repeater" ref="foreachSplitDoc"/>

                    </sequence>
                </sequence>
            </process>
```

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing

IBM Corporation

North Castle Drive

Armonk, NY 10504-1785

U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing

Legal and Intellectual Property Law

IBM Japan Ltd.

19-21, Nihonbashi-Hakozakicho, Chuo-ku

Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be

incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation

J46A/G4

555 Bailey Avenue

San Jose, CA 95141-1003

U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© IBM 2015. Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. 2015.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at http://www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java™ and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Linear Tape-Open, LTO, the LTO Logo, Ultrium and the Ultrium Logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

Connect Control Center®, Connect:Direct®, Connect:Enterprise®, Gentran®, Gentran®:Basic®, Gentran:Control®, Gentran:Director®, Gentran:Plus®, Gentran:Realtime®, Gentran:Server®, Gentran:Viewpoint®, Sterling Commerce™, Sterling Information Broker®, and Sterling Integrator® are trademarks or registered trademarks of Sterling Commerce®, Inc., an IBM Company.

Other company, product, and service names may be trademarks or service marks of others.

Index

A
adapters, bootstrap   2

B
batch processing   4
batch transactions   2
bootstrap adapters   2
BP Metadata Info service   5
BPML All activity   2
business process
   and
      batch transactions   2
      bootstrap adapters   2
      retiring services   3
   design reusability   1
   excess activities in   3
   improving processing times   3
   performance, considerations   3
   predefined, using   2
   scheduling   3
   subprocesses   2

D
deferred enveloping   4
Deferred mode   4

E
EDI enveloping, deferred mode   4
enveloping, deferred mode   4
execution speed   5

F
file system, limit calls to   6

G
Get Document Info service   5

I
inline invoke   5
invoke suprocesses
   asynchronously   5
   synchronously   5

O
onFault activity   5

P
performance, optimizing in process
  models   3

persistence level
   business process   4
   freeing system resources   4
predefined business process, use
  recommendation   2

R
retiring services   3

S
schedule, business process   3
Sleep service   4
subprocess, managing size   2

W
Wait service   4

X
XPath search, specify path   5

IBM ®

Product Number:

Printed in USA